

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**DOĞA BİLİMLERİ VE MÜHENDİSLİK FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ PROGRAMI**

**ÖDEV KONUSU**

**GEOMETRİK PROBLEMLER**

**HAZIRLAYAN**

**ÖĞRENCİ ADI: Aybüke TÜRİDİ**

**Elif Beyza BEYAZ**

**ÖĞRENCİ NUMARASI: 220502005**

**220502033**

**DERS SORUMLUSU**

**Prof. Dr. H. Tarık DURU**

**TARİH**

**2.01.20234**

**İÇİNDEKİLER**

[1. ÖZET (ABSTRACT) 3](#_Toc136267779)

[2. GİRİŞ (INTRODUCTION) 3](#_Toc136267780)

[3. YÖNTEM (METHOD) 3](#_Toc136267781)

[4. SONUÇ VE ÖĞRENİLEN DERSLER 3](#_Toc136267784)

[5. KAYNAKÇA 3](#_Toc136267785)

6. GİTHUB LİNKLERİ………………………………………………………………………………

1. **ÖZET**

Bu C++ programında, Nokta, DogruParcasi, Daire ve Ucgen sınıfları oluşturularak geometrik problemleri çözmek üzere bir yapı geliştirilmiştir. Nokta sınıfı, bir noktanın x ve y koordinatlarını yönetirken, DogruParcasi sınıfı iki noktayı içeren bir doğru parçasını temsil eder. Daire sınıfı, merkezi ve yarıçapıyla bir daireyi modellerken, Ucgen sınıfı üç noktayı içeren bir üçgenin özelliklerini taşır. Her sınıfın özel metotları, geometrik hesaplamalar ve temsiller için kullanılmıştır.

1. **GİRİŞ**

Bu program, temel geometrik problemleri çözmek amacıyla oluşturulan Nokta, DogruParcasi, Daire ve Ucgen sınıflarını içermektedir. Nokta sınıfı, koordinatları yönetirken, DogruParcasi sınıfı doğru parçalarını temsil eder. Daire sınıfı dairelerin özelliklerini modellemekte ve Ucgen sınıfı üçgenlerin özelliklerini taşımaktadır.

Bu kodlar C++ programlama dilinde yazılmış ve geometrik şekilleri temsil eden sınıfları içermektedir. Program, Nokta, DogruParcasi, Daire ve Ucgen sınıflarını tanımlamakta ve bu sınıfların özelliklerini ve davranışlarını içermektedir. Ayrıca, bu sınıfların birbirleriyle etkileşimini gösteren bir test programı içermektedir.

Program, üç ana geometrik şekil sınıfını içerir: **Nokta**, **DogruParcasi**, **Daire** ve **Ucgen**.

**Nokta** sınıfı, iki boyutlu bir noktanın koordinatlarını temsil eder ve bu nokta üzerinde temel operasyonları gerçekleştirebilmek için getter, setter ve diğer yardımcı fonksiyonları içerir.

**DogruParcasi** sınıfı, iki noktayı birleştirerek bir doğru parçası oluşturur. Ayrıca, doğru parçasının uzunluğunu, kesişim noktasını ve orta noktasını hesaplamak için gerekli işlevleri ve fonksiyonları içerir.

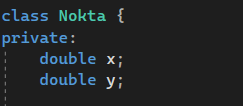
**Daire** sınıfı, bir merkez noktası ve bir yarıçapı temsil eder ve dairenin alanını, çevresini ve başka bir daire ile kesişip kesişmediğini kontrol etmek için işlevleri içerir.

**Ucgen** sınıfı, üç noktayı kullanarak bir üçgen oluşturur ve üçgenin alanını, çevresini ve üçgenin özelliklerini içeren işlevlere sahiptir.

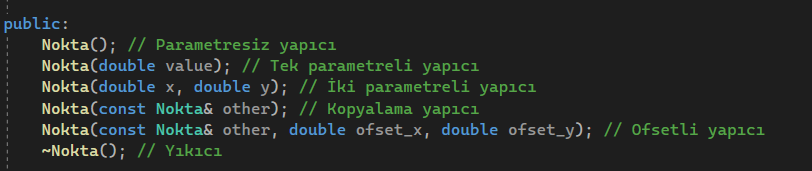
1. **YÖNTEM**
2. **Nokta**

İlk sınıfımız diğer sınıfların gereklilikleri sebebiyle nokta sınıfıdır.

Header dosyası:

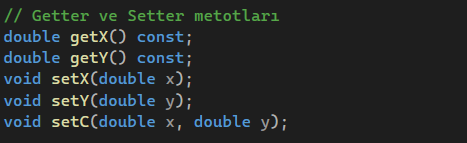


**X** ve **y**, Nokta sınıfının özel (private) veri üyeleridir ve bir noktanın x ve y koordinatlarını saklar.



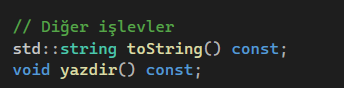
Bu fonksiyonlar, Nokta sınıfının farklı yapıcı fonksiyonlarını tanımlar.

Kod parçasının son satırı nesne yok edildiğinde yani bellek boşaltıldığında çağırılmış olur.



Bu fonksiyonlar, "Nokta" sınıfının veri üyelerine güvenli bir erişim sağlar ve sınıfın dışındaki kodun bu değerleri kontrol etmesine ve değiştirmesine izin verir.

**const** anahtar kelimesi, sınıfın içyapısını gizleyerek ve sınıfın dışındaki etkileşimleri kontrol ederek sınıfın daha güvenli ve modüler olmasına yardımcı olur.



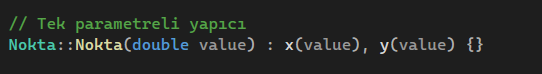
Bu kod parçası ifadeleri metine dönüştürerek yazdırmayı sağlar.

Cpp dosyası için 5 adet yapıcı bulunur.

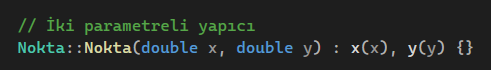


Bu yapıcı fonksiyon, nesne oluşturulurken hiçbir parametre almayan durumu ele alır.

X ve y değerleri sıfır olarak atanır.



Bu yapıcı fonksiyon, değerleri tek double türüne atar.



Bu yapıcı fonksiyon, değerleri iki double türüne parametreye alır.



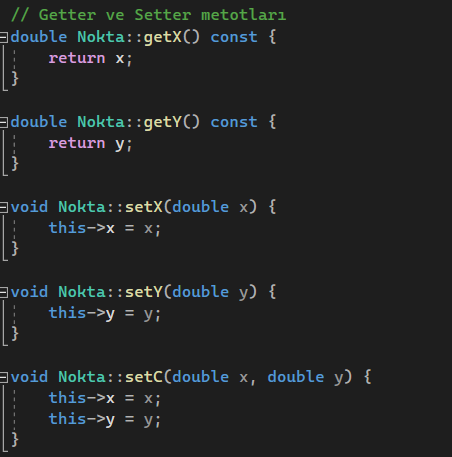
Bu yapıcı fonksiyon, bir "Nokta" nesnesinin kopyasını oluşturmak için kullanılır.



Bu yapıcı fonksiyon, başka bir **Nokta** nesnesinin belirli bir ofsetle (kaydırma) oluşturulmuş kopyasını yapmak için kullanılır. Yani, bu yapıcı fonksiyon, mevcut bir **Nokta** nesnesini alır, bu noktanın x ve y koordinatlarını belirtilen ofset değerleri kadar değiştirerek yeni bir **Nokta** nesnesi oluşturur.



Bu fonksiyon, nesne bellekten silindiğinde çağrılır.

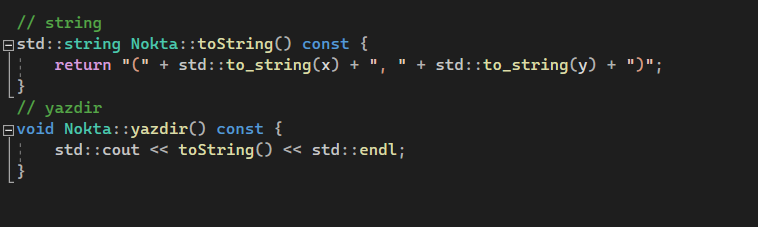


**get** metodu, bir "Nokta" nesnesinin koordinatını döndürür.

**const** niteliği ile işaretlenmiş, bu da bu metodun sınıfın içinde çağrıldığında nesnenin durumunu değiştirmeyeceği anlamına gelir.

**set** metodu, bir "Nokta" nesnesinin koordinatını belirtilen bir değerle değiştirir.

this ile atama yapılır.

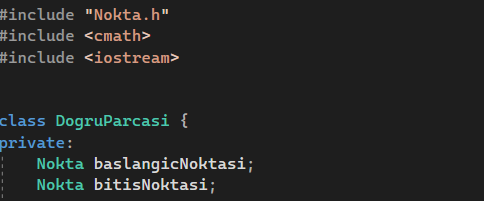


String veri tipine döndürme ve ekrana yazdırma fonksiyonları tanımlanır.

**toString()** ile çevirme yapılır **cout** ile ekrana yazdırma gerçekleşir.

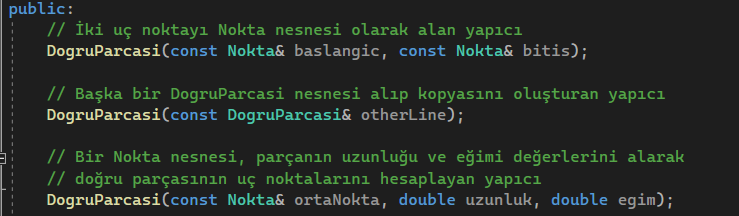
1. **Doğru Parçası**

**DogruParcasi** sınıfı, iki uç noktayı **Nokta** sınıfının nesneleri olarak içerir. Bu noktalar, **baslangicNoktasi** ve **bitisNoktasi** adında özel veri üyeleri aracılığıyla temsil edilir.



Bu nedenle nokta sınıfının header dosyası bu kodlamaya dâhil edilmiştir.

Matematiksel hesaplamalar için **cmath** kütüphanesine de ihtiyaç duyulur.



3 adet yapıcı fonksiyon bulunur.

İki Nokta ile Yapıcı:

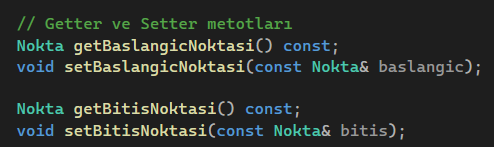
Bu yapıcı, başlangıç ve bitiş noktalarını parametre olarak alır ve doğru parçasını oluşturur.

Kopyalama Yapıcı:

Bu yapıcı, başka bir DogruParcasi nesnesini alır ve onun kopyasını oluşturur.

Orta Nokta, Uzunluk ve Eğim ile Yapıcı:

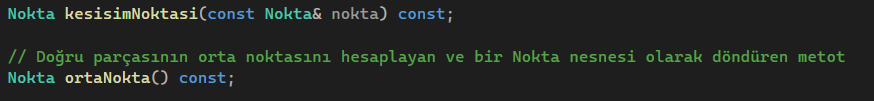
Bu yapıcı, bir orta nokta, uzunluk ve eğim değerlerini alarak doğru parçasının uç noktalarını hesaplar.



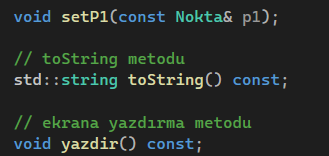
Get ve set metotları ile noktalar yerleştirilir. Const ile veri gizliliği sağlanır.



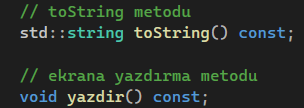
Uzunluk hesaplanır ve double veri tipine atanır.



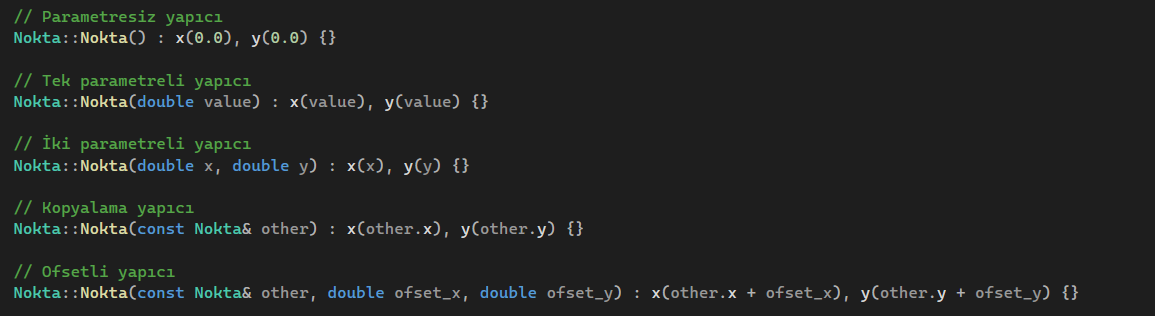
Bu iki fonksiyon, **DogruParcasi** sınıfının özel işlevlerini gerçekleştiren üye fonksiyonlardır. İki fonksiyon da **const** niteliklidir, bu da bu fonksiyonların sınıfın durumunu değiştirmeyeceği ve sadece okuma işlemleri gerçekleştireceği anlamına gelir.



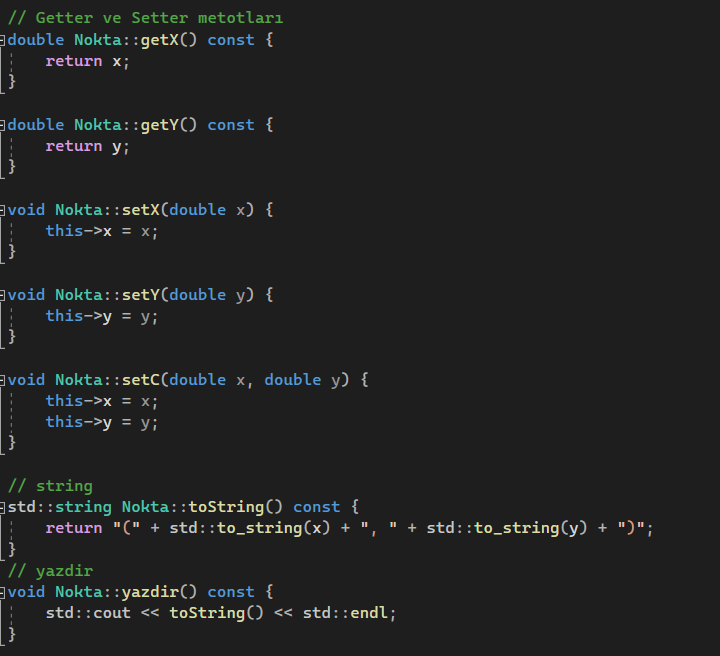
Bu fonksiyon, doğru parçasının başlangıç noktasını (**baslangicNoktasi**) belirlemek için kullanılır. Fonksiyon, **p1** adlı bir **Nokta** nesnesini parametre olarak alır ve bu noktayı doğru parçasının başlangıç noktası olarak ayarlar.



String halinde yazdıran fonksiyonlar çağırılır.



4 adet özel nitelikli yapıcı fonksiyonlar bulunur.



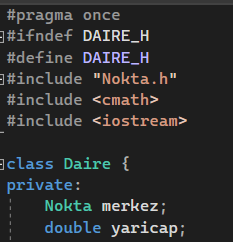
**get** metodu, bir "Nokta" nesnesinin koordinatını döndürür.

const niteliği ile işaretlenmiş, bu da bu metodun sınıfın içinde çağrıldığında nesnenin durumunu değiştirmeyeceği anlamına gelir.

**set** metodu, bir "Nokta" nesnesinin koordinatını belirtilen bir değerle değiştirir.

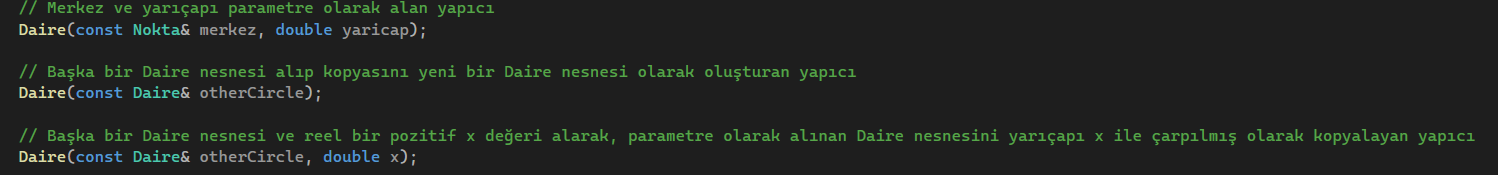
this ile atama yapılır ve yazdırır.

1. **Daire**



Daire sınıfında nokta sınıfı ve math kütüphanesi dâhil edilir.

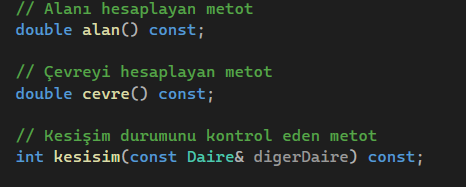
Kullanılacak değerler private olarak belirlenir. (merkez, yarıçap)



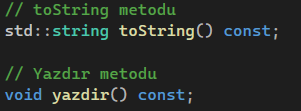
İlk yapıcı fonksiyon, bir merkez noktası ve bir yarıçap alarak bir daire oluşturur. Bu şekilde bir daire, belirli bir merkezde odaklanmış ve belirli bir yarıçapa sahip olacaktır.

Diğer iki yapıcı fonksiyon kopya oluşturur.

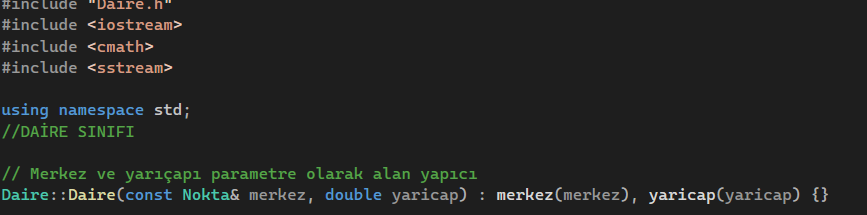
Biri mevcut dairenin kopyasını oluştururken diğeri x faktörü etkileşimiyle oluşan dairenin kopyasını oluşturur.



Alanı ve çevreyi hesaplaya fonksiyonlar double veri tipinde çağırılır. Keşişimleri de kopyalanarak çağırılır.



Yazdıran fonksiyon çağırılır.



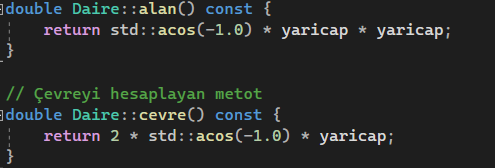
Bu kısım, **Daire** sınıfının tanımı ve yapıcı fonksiyonunun implementasyonu. **Daire** sınıfı, bir **Nokta** nesnesi ve bir **double** yarıçapı alarak bir daire oluşturur.



**Daire** nesnesini alarak kopyasını oluşturan yapıcı fonksiyon.

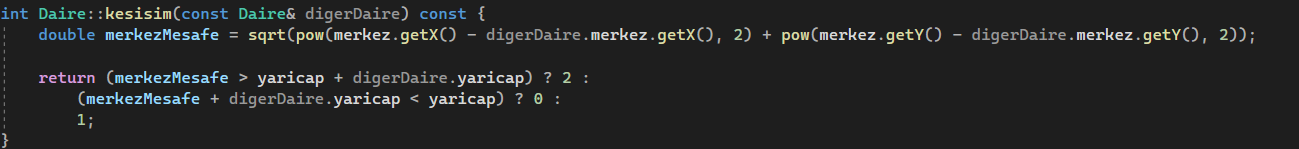


Bu kısım, başka bir **Daire** nesnesini ve bir reel sayı olan **x** değerini alarak, **otherCircle**'ın yarıçapını **x** ile çarparak yeni bir **Daire** nesnesi oluşturan yapıcı fonksiyon.

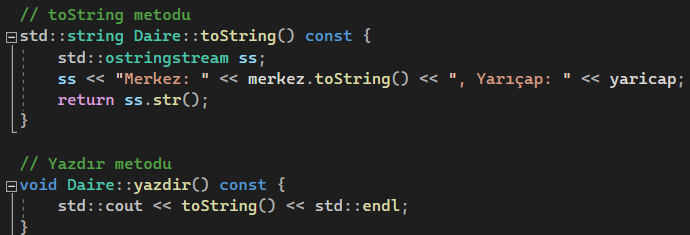


Alan, π (pi) sayısının yarıçapın karesi ile çarpılmasıyla bulunur. Pi sayısı acos(-1,0) ile sağlanır.

Çevre, π (pi) sayısının iki katı ile yarıçapın çarpılmasıyla bulunur.

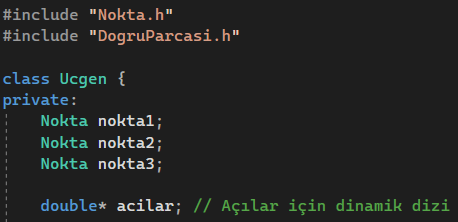


Bu kısım, iki dairenin kesişim durumunu kontrol eden metot. Eğer iki daire birbirine değmiyorsa **2**, iç içe geçmişlerse **0**, aksi takdirde **1** değeri döndürür.



Dairenin bilgilerini bir string olarak döndüren ve yazdıran fonksiyonlar.

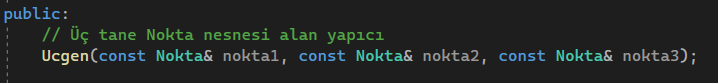
1. **Üçgen**



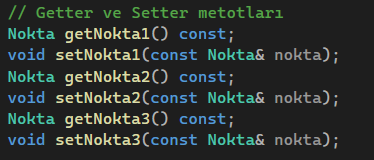
Üçgeni hesaplarken nokta ve doğru parçası sınıfları kullanılır.

**Üçgen** sınıfı, üç adet **Nokta** nesnesini alarak üçgenin üç köşe noktasını temsil eder.

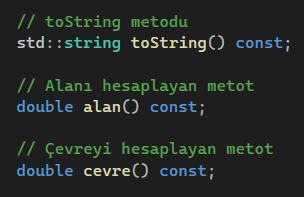
**double\* acilar;** ifadesi, üçgenin iç açılarını tutmak için bir dinamik dizi tanımlar.



Bu yapıcı fonksiyon, üç nokta alır ve bu noktaları kullanarak bir üçgen oluşturur. Noktalar, **Nokta** sınıfından gelir.



Bu metotlar, sınıfın özel üye değişkenlerine (noktalar) erişim sağlar. **Getter** metotları, özel üye değişkenlerin değerlerini döndürür, **Setter** metotları ise bu değerleri ayarlar. Bu işlem üçgenden kaynaklı olarak 3 kere yapılır.



Bu metot, sınıfın bilgilerini içeren bir string döndürür. Örneğin, üç noktanın koordinatlarını içerebilir. Alan ve çevre hesaplayan fonksiyonlar çağırılır.

1. **SONUÇ VE ÖĞRENİLEN DERSLER**

Bu ödev, C++ programlama dilinde nesne tabanlı programlama konseptlerini etkili bir biçimde uygulamayı amaçlayan bir proje olarak gerçekleştirilmiştir. Tasarlanan Nokta, DogruParcasi, Daire ve Ucgen sınıfları, geometrik nesnelerin temsilini sağlamakla kalmamış, aynı zamanda bu nesneler arasındaki ilişkileri ve özellikleri etkili bir şekilde yönetebilecek metotları içermektedir. Nesne tabanlı programlama prensipleri sayesinde modüler bir yapı oluşturulmuş ve her sınıfın kendi sorumluluk alanında bağımsız çalışması sağlanmıştır.

Bu proje, sınıf ve nesne kavramlarıyla birlikte, kalıtım, çok biçimlilik ve sınıflar arası etkileşim gibi Nesne Yönelimli Programlama prensiplerini pratiğe dökme imkânı sağlamıştır. Geometrik problemlerin soyutlanması ve uygun sınıfların tasarlanması, gerçek dünya senaryolarında nesne tabanlı programlamanın gücünü kavramak açısından değerli bir deneyim sunmuştur.

1. **KAYNAKÇA**

Geçerli değere sahip kaynak bulunmamaktadır.

1. **GİTHUB LİNKLERİ**

Aybüke TÜRİDİ

<https://github.com/220502005>

Elif Beyza BEYAZ

<https://github.com/beyzabeyaz0>